
Major: ECE

Student No. U2301068, U2301108

Chinese name: 佟一， 蔡丰锴

Name in pinyin：Tong Yi, Cai Fengkai

# Content

EES4408 Project: Wordle ..... 3

1. Problems ..... 3
2. Connection to class ..... 3
3. Data ..... 4
4. Classifiers ..... 5
5. Clustering ..... 6
6. Question about word “EERIE” ..... 7

# EES4408 Project: Wordle

## 1. Problems


"Wordle" is a popular daily puzzle game, where players try to guess a five-letter word in six or fewer tries with feedback given after each guess. The game can be played in regular or hard mode, where the latter requires players to use correct letters in subsequent guesses. The color of the tiles changes after each guess to indicate whether the letter is in the word and in the correct location (green), in the word but in the wrong location (yellow), or not in the word at all (gray). Fig. 1 is an example solution where the correct result was found in three tries.

Fig. 1

The data set contains information about the solution words, the number of players, the percentage of players who successfully guessed the word in one to six tries or were unable to solve it, and whether they played in regular mode or hard mode.

**Problems:**

a) Develop a model to predict the distribution of reported results (1, 2, 3, 4, 5, 6, 7 or more tries (X)) for the solution word.

b) Develop a model to classify solution words by difficulty and use this model to identify the difficulty of the word EERIE.

## 2. Connection to class

**a) Correlation**

Use the correlation map to show the relationship between each feature.

**b) Multi-output classifiers: linear regression, random forest**

For problem a), different classifiers are used to do the prediction. Then, compare the mean absolute error (MAE) between them.

**c) Clustering: K-means, Agglomerative (top up)**

For problem b), do the clustering for difficulty identification.

**d) PCA**

Make dimensional reduction by PCA so that can visualize the clustering results.

## 3. Data

The data we used was after cleaning and feature extraction. Then, each word has the following attributes:

**Average tries:** A weighted average is performed based on the distribution of the number of tries, and X is considered as 8 for the calculation to obtain the expected value.

**Hard mode percentage:** The percentage of people who choose the hard mode.

**The highly used letter:** Whether letters that appear in high frequency in words such as *e, t, a, o, i, n* appear or not. Noted them as "e12, t9, a8, o7, i7, n7", respectively.

**Number of vowels:** The number of vowel letters in a word. Noted as "vowel num".

**Number of duplications:** The number of duplicated letters in a word. Noted as "dup num".

**Letter encoding:** Encode the words by encoding each letter as an integer from 1 to 26.

Use the correlation map to show the relationship between each feature (Fig. 2).

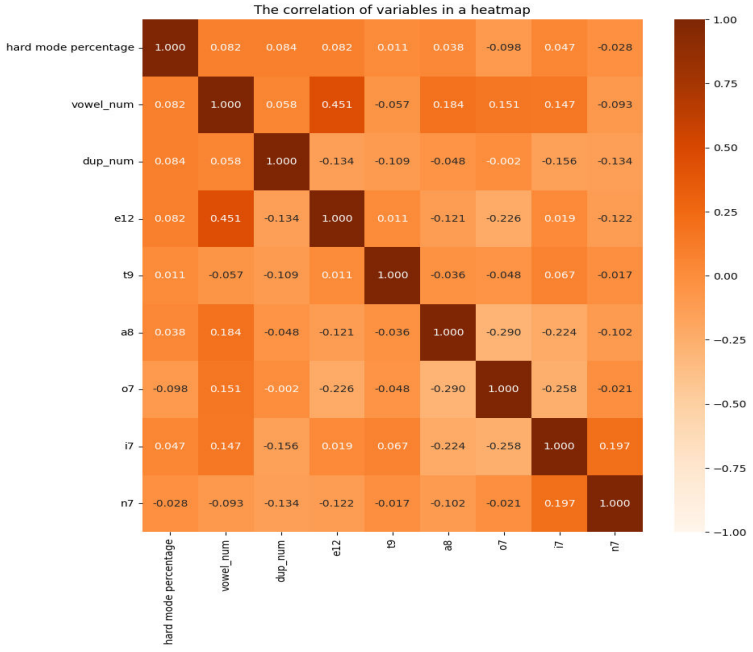


Fig. 2

| | Vowel | Duplicate | e | t | a | o | i | n |
|---|---|---|---|---|---|---|---|---|
| Hard mode pct | **0.082** | **0.084** | **0.082** | 0.011 | 0.038 | **-0.098** | 0.047 | -0.028 |

It can be observed in the Table that the number of vowel letters, the number of duplicated letters, the letter e, and the letter o are much more significant to the hard mode percentage.

As for learning, split the training set and the testing set randomly, the testing set accounts for 10%.

## 4. Classifiers

RegressorChain is applied for the multi-input multi-output model.

### a) Linear regression

Linear regression:
Mean absolute error of training and testing set

| | 1 try | 2 tries | 3 tries | 4 tries | 5 tries | 6 tries | 7 or more tries (X) |
|---|---|---|---|---|---|---|---|
| **Train** | 0.398949 | 2.235824 | 5.023529 | 3.850355 | 3.848730 | 4.170287 | 2.201394 |
| **Test** | 0.537537 | 2.816582 | 5.903289 | 3.282952 | 4.279671 | 4.706402 | 2.342714 |

### b) Random forest

Random forest:
Mean absolute error of training and testing set

Depth: 10, 500 estimators

| | 1 try | 2 tries | 3 tries | 4 tries | 5 tries | 6 tries | 7 or more tries (X) |
|---|---|---|---|---|---|---|---|
| **Train** | 0.161477 | 0.850565 | 2.658122 | 2.526925 | 2.158641 | 2.546484 | 1.312802 |
| **Test** | 0.420176 | 2.579491 | 4.895555 | 3.746755 | 4.077296 | 4.138264 | 1.693424 |

### c) Comparison

Overall, it can be seen that the random forest performs better than the linear regression does, more obvious in the training set.

Linear regression assumes a linear relationship between the input features and the target variable. Random forest can capture nonlinear relationships between multiple input variables and multiple output variables effectively. Each decision tree in the forest can learn different aspects of the relationship.

Thus, random forest tends to perform well due to its ability to handle complex interactions without the assumption of linearity.

### d) Tuning the parameters in random forest

In training set, different depths. (500 estimators)

| | 1 try | 2 tries | 3 tries | 4 tries | 5 tries | 6 tries | 7 or more tries (X) |
|---|---|---|---|---|---|---|---|
| **5** | 0.259762 | 1.830526 | 4.717314 | 3.773708 | 3.554200 | 4.046390 | 1.712167 |
| **10** | 0.161477 | 0.850565 | 2.658122 | 2.526925 | 2.158641 | 2.546484 | 1.312802 |
| **15** | 0.147061 | 0.794792 | 2.480242 | 2.264113 | 2.050390 | 2.367218 | 1.210151 |

The first one is the depth of the trees in the forest, fix the number of estimators and evaluate the depth. Increasing the depth of the trees allows the model to capture more complex relationships in the data that may better fit the training data.

As shown in the Table above, the error in the training set decreases as the tree becomes deeper.

In testing set, different depths. (500 estimators)

| | 1 try | 2 tries | 3 tries | 4 tries | 5 tries | 6 tries | 7 or more tries (X) |
|---|---|---|---|---|---|---|---|
| **5** | 0.414610 | 2.876271 | 5.458633 | 3.558455 | 4.732272 | 4.373281 | 1.718370 |
| **10** | 0.420176 | 2.579491 | 4.895555 | 3.746755 | 4.077296 | 4.138264 | 1.693424 |
| **15** | 0.424305 | 2.581407 | 5.023807 | 3.774962 | 4.149968 | 4.217389 | 1.710278 |

However, deeper trees can increase the risk of overfitting. Since the model learns the data too well, resulting in poor generalization to unseen data in the testing set.

As shown in the Table above, the error reduces when increasing the depth from 5 to 10, and regrows when the depth goes to 15.

In training set, different estimators. (depth: 10)

| | 1 try | 2 tries | 3 tries | 4 tries | 5 tries | 6 tries | 7 or more tries (X) |
|---|---|---|---|---|---|---|---|
| **50** | 0.164800 | 1.015148 | 2.799619 | 2.673849 | 2.229937 | 2.664442 | 1.290570 |
| **300** | 0.160306 | 0.852335 | 2.721504 | 2.523656 | 2.169069 | 2.603983 | 1.289896 |
| **500** | 0.161477 | 0.850565 | 2.658122 | 2.526925 | 2.158641 | 2.546484 | 1.312802 |
| **800** | 0.160930 | 0.886972 | 2.701088 | 2.528144 | 2.166266 | 2.576335 | 1.286042 |

In testing set, different estimators. (depth: 10)

| | 1 try | 2 tries | 3 tries | 4 tries | 5 tries | 6 tries | 7 or more tries (X) |
|---|---|---|---|---|---|---|---|
| **50** | 0.413767 | 2.859294 | 5.182912 | 4.085274 | 4.471576 | 4.292732 | 1.737660 |
| **300** | 0.425812 | 2.637768 | 5.106645 | 3.894325 | 4.276358 | 4.303675 | 1.729026 |
| **500** | 0.420176 | 2.579491 | 4.895555 | 3.746755 | 4.077296 | 4.138264 | 1.693424 |
| **800** | 0.421124 | 2.571727 | 5.027800 | 3.742254 | 4.107128 | 4.252042 | 1.703398 |

The other parameter is the number of estimators. Increasing the number of estimators can lead to better performance. This is because the predictions from multiple trees are averaged, so to reduce the variance. Thus, making the model more robust, both for the training set and the testing set.

As shown in the Tables above, the error reduces for both the training set and testing set when the number of estimators increases.

However, adding too many trees beyond a certain point might not significantly improve performance but will increase computational cost.

## 5. Clustering

To obtain the number of clusters, we calculate the sum of square error with respect to the cluster number according to the elbow method and take the *n* value at the “elbow” point as the number of clusters.

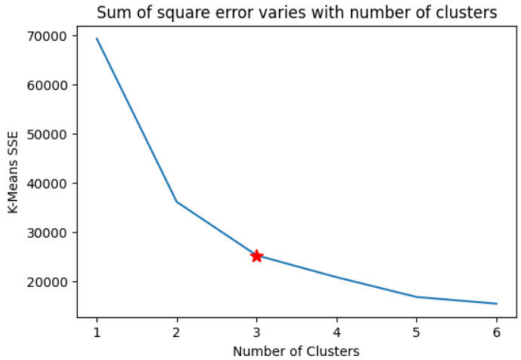


Fig. 3

As shown in Fig. 3, visually pick 3 clusters at the elbow. Take the average trying times, and sort from small to large, namely [Easy, Medium, Hard].

Firstly, using PCA to do dimensional reduction to visualize the results intuitively. The eigenvalues are shown below and pick the two largest as the principal components.

```
Eigenvalues: [139.2  42.1   9.8   3.4   1.4   0.4   0.3   0. ]
```

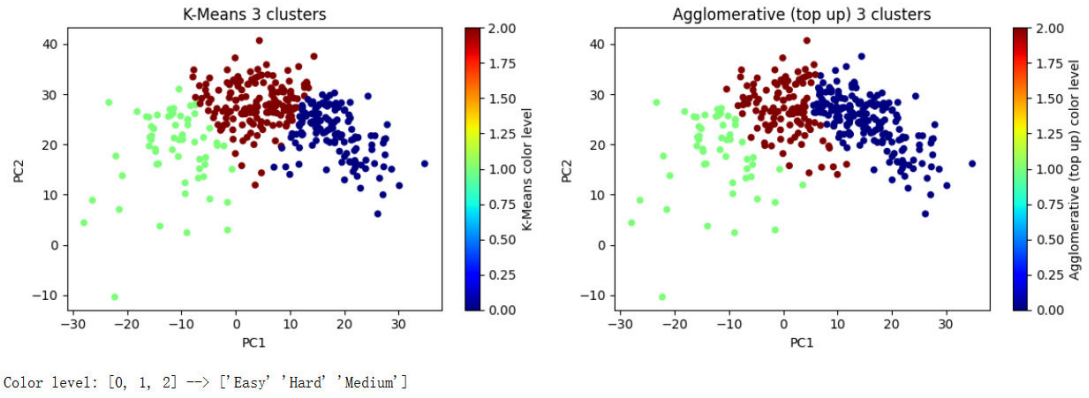


Fig. 4

As shown in Fig. 4, The results of K-Means and Agglomerative (top up) (using the variance in each cluster to measure the distance) are almost the same, because the data is compact and the number of the clusters is prespecified.

## 6. Question about word “EERIE”

First, apply the random forest model with the depth of 10 and 500 estimators. Get the predicted distribution of the trying result. Then, adapt to the K-Means clustering result and obtain the difficulty level.

| | 1 try | 2 tries | 3 tries | 4 tries | 5 tries | 6 tries | 7 or more tries (X) | hard mode percentage |
|---|---|---|---|---|---|---|---|---|
| 0 | 0.986 | 4.330629 | 17.983806 | 30.404184 | 26.850732 | 16.28346 | 3.563967 | 0.01 |

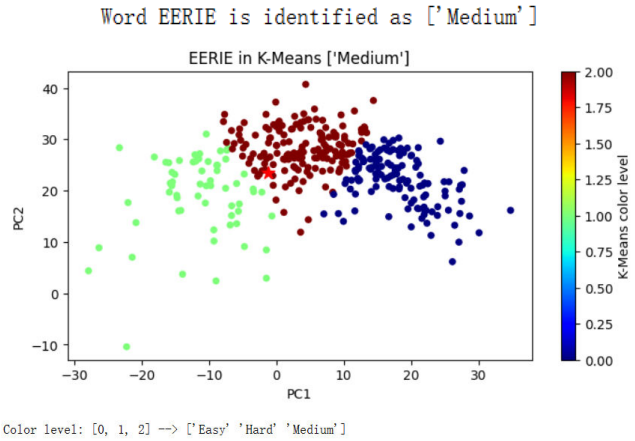


Fig. 5

As shown in Fig. 5, the word EERIE (marked as the star) falls in the region of “Medium”, which represents the difficulty level of word EERIR.